**Wdrożenie APM dla Node.js (z Secret Token)**

**0) Założenia**

* Masz: **APM Server URL** (np. https://apm.twojadomena:8200) i **Secret Token**.
* **Nie używamy API keya.**
* **User/Password do Kibany/ES nie są używane przez agenta** (przydadzą się tylko do pracy w Kibanie/ES).

**1) Instalacja pakietu agenta**

npm i elastic-apm-node --save

# lub

yarn add elastic-apm-node

**2) Konfiguracja przez zmienne środowiskowe (zalecane)**

Ustaw w .env / sekretach CI:

ELASTIC\_APM\_SERVICE\_NAME=node-api

ELASTIC\_APM\_SERVER\_URL=https://apm.twojadomena:8200

ELASTIC\_APM\_SECRET\_TOKEN=xxxxxxxxxxxxxxxx

ELASTIC\_APM\_ENVIRONMENT=production

# Prywatność/higiena danych:

ELASTIC\_APM\_CAPTURE\_BODY=off # off|errors|transactions|all (zostaw off)

ELASTIC\_APM\_VERIFY\_SERVER\_CERT=true # w produkcji true

# Jeśli APM Server ma własne CA (self-signed):

# ELASTIC\_APM\_SERVER\_CA\_CERT\_FILE=/etc/ssl/certs/apm-ca.pem

**Ważne:** Nie ustawiaj ELASTIC\_APM\_API\_KEY — używamy wyłącznie SECRET\_TOKEN.

**3) Start agenta NAJWCZEŚNIEJ w procesie**

**A) CommonJS**

**apm.js**

// Ładuj JAK NAJWCZEŚNIEJ – przed Express/Koa itp.

require('elastic-apm-node').start({

serviceName: process.env.ELASTIC\_APM\_SERVICE\_NAME || 'node-api',

serverUrl: process.env.ELASTIC\_APM\_SERVER\_URL,

secretToken: process.env.ELASTIC\_APM\_SECRET\_TOKEN, // tylko Secret Token

environment: process.env.ELASTIC\_APM\_ENVIRONMENT || 'development',

// RODO / ochrona danych

captureBody: process.env.ELASTIC\_APM\_CAPTURE\_BODY || 'off',

captureHeaders: true,

sanitizeFieldNames: [

'\*password\*','\*token\*','authorization','set-cookie','\*secret\*','\*credit\*'

],

// TLS/self-signed

serverCaCertFile: process.env.ELASTIC\_APM\_SERVER\_CA\_CERT\_FILE,

verifyServerCert: String(process.env.ELASTIC\_APM\_VERIFY\_SERVER\_CERT) !== 'false'

});

**server.js (główny plik)**

require('./apm'); // MUSI być pierwsze

const express = require('express');

const app = express();

app.get('/health', (req, res) => res.send('ok'));

// Test błędu → powinien pojawić się w APM (Errors):

app.get('/\_apm/test-error', () => { throw new Error('Boom'); });

app.listen(3000, () => console.log('API listening on 3000'));

**B) ESM (type: "module")**

Najprościej pre-loadem:

node -r elastic-apm-node/start server.mjs

(albo **pierwszy import** w server.mjs: import 'elastic-apm-node/start'), konfiguracja z ENV jak wyżej.

**4) Uruchomienie i weryfikacja**

1. Uruchom aplikację z ustawionymi ENV.
2. Odwiedź kilka swoich endpointów, wywołaj /\_apm/test-error.
3. W Kibanie wejdź w **Observability → APM → Services** i sprawdź, czy widzisz node-api, transakcje i błędy.

**5) Dobre praktyki (prod)**

* **Tylko Secret Token**: nie mieszaj z API key.
* **TLS**: zostaw verifyServerCert=true; przy self-signed podaj ścieżkę do CA zamiast wyłączać weryfikację.
* **Prywatność**: trzymaj capture\_body=off (ew. errors); aktualizuj sanitizeFieldNames.
* **Sampling** (duży ruch): zacznij od transaction\_sample\_rate=0.2 (ustawisz centralnie w APM Settings).
* **Custom spans** (opcjonalnie):
* const apm = require('elastic-apm-node');
* app.get('/work', async (req, res) => {
* const span = apm.startSpan('heavy-calc');
* try {
* // ...praca
* } finally {
* if (span) span.end();
* }
* res.send('done');
* });

**TL;DR (checklista)**

1. npm i elastic-apm-node
2. ENV: SERVER\_URL, SECRET\_TOKEN, SERVICE\_NAME, ENVIRONMENT (+ ewentualnie SERVER\_CA\_CERT\_FILE)
3. Załaduj agenta **przed** frameworkiem (require('./apm') lub -r elastic-apm-node/start)
4. Uderz endpointy + /\_apm/test-error
5. Sprawdź **Observability → APM → Services** w Kibanie